DETEKSI DAN KOREKSI KESALAHAN INFORMASI DALAM SANDI BINER DENGAN MENGGUNAKAN METODE HAMMING

Wiwik Anggraeni
Program Studi Sistem Informasi
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. + 62 31 5939214, Fax. + 62 31 5913804
Email : wiwik@its-sby.edu

ABSTRAK

Seiring semakin berkembangnya teknologi komunikasi semakin meningkat pula pelayanan dibidang informasi yang menuntut penyampaian informasi yang lebih sempurna. Kesempurnaan suatu informasi bisa meliputi kecepatan pengiriman, tidak adanya informasi yang hilang atau rusak, pengamanan, dan lain sebagainya.

Dalam pengiriman informasi kadang-kadang sering terjadi kesalahan informasi yang diterima dari pengirim informasi itu sendiri. Kesalahan itu dapat diakibatkan oleh gangguan dari media transmisinya ataupun faktor-faktor lain. Perangkat komunikasi data dituntut mampu menangani masalah tersebut. Salah satu kemampuan yang diharapkan dari perangkat komunikasi data adalah dapat melakukan deteksi atau koreksi kesalahn informasi.

Salah satu metode yang sering digunakan untuk mendeteksi kesalahan adalah metode hamming. Metode hamming tersebut tidak hanya mampu untuk mendeteksi kesalahan saja akan tetapi mempunyai kemampuan juga untuk mengoreksi kesalahan informasi secara otomatis tanpa harus mengirimkan informasi itu kembali.

Kata kunci : Metode *Hamming*, deteksi kesalahan informasi, koreksi kesalahan informasi

1. PENDAHULUAN

1.1 Latar Belakang

Dalam transmisi data yang menjangkau lebih dari ratusan dan bahkan ribuan kilometer, timbulnya error merupakan sesuatu yang tidak dapat dihindarkan. Dalam suatu transmisi yang mutahirpun gangguan-gangguan pada saluran tersebut tidak dapat diatasi secara sempurna. Gangguan-gangguan tersebut akan dapat menyebabkan data yang diterima berbeda dengan yang dikirim.

Pada sistem komunikasi penggunaan sistem digital merupakan pilihan terbaik,terutama untuk komunikasi jarak jauh, karenanya diperlukan sistem yang baik untuk menyatakan isyarat analog menjadi isyarat digital yang berupa sandi biner. Persoalannya adalah bagaimana menghasilkan sandi digital yang dapat dikirimkan tanpa mengalami kesalahan atau meminimalkan kesalahan sehingga pesan yang diterima sesuai dengan yang dikirimkan.

1.2 Rumusan Permasalahan

Seperti pada umumnya bila diketemukan kesalahan pada data yang diterima maka pengirim diminta untuk mengirimkan kembali data tersebut. Jelas cara ini akan memakan waktu dan biaya yang tidak sedikit. Oleh karena itu dicari suatu cara yang lebih efisien yaitu dengan menggunakan metode koreksi yang dilakukan secara otomatis dengan penggunaan sandi tertentu. Sandi itu mengandung cukup redundansi yang memungkinkan terdeteksinya kesalahan sekaligus mengoreksinya sehingga tidak diperlukan transmisi ulang.

1.3 Batasan Masalah

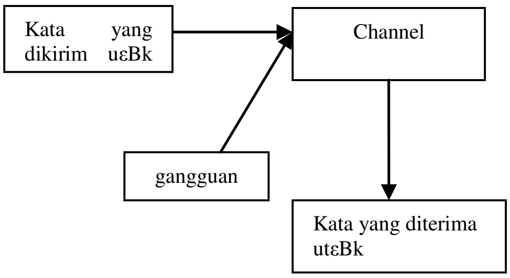
Adapun batasan yang digunakan adalah

1. Hanya dapat mendeteksi dan mengoreksi kesalahan pada data tersandi 7 bit
2. Bila terjadi kesalahan maka jumlah bit yang dapat dideteksi maupun dikoreksi hanya 1 per karakter.
3. Hanya dipusatkan pada kode-kode linier.
4. Pembahasan hanya dilakukan pada kode kode sistematis, yaitu bit bit cek paritasnya selalu muncul di akhir kata kode menyusul urutan data biner orisinilnya.

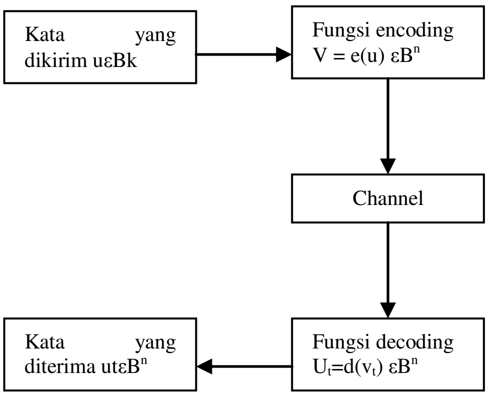
2. PENGKODEAN

a. Encoding

Proses pengiriman berita dapat dilihat di gambar berikut :



Fungsi satu-satu $e:B^k \rightarrow B^n$, $k \leq n$ disebut sebagai fungsu encoding(n,k). Jika $u \; \varepsilon \; B^k$ maka e(u) $\varepsilon \; B^n$ disebut kata kode dari u. Sedangkan untuk proses pengiriman kata kode nya sebagai berikut :



b. Decoding

Fungsi pada $d : B^n \rightarrow B^k$ disebut sebagai fungsi decoding(n,k) yang berpasangan dengan fungsi encoding $e:B^k \rightarrow B^n$, jika d(vt) = u $\varepsilon \; B^k$ dan ketika channel tidak mempunyai gangguan mengakibatkan u = u.

c. Tipe-tipe Kode

Ada 2 jenis tipe kode yang sering dipakai, yaitu :

1. Kode blok
   Misalkan B = {0,1} dan fungsi encoding $e:B^k \rightarrow B^n$ maka jumlah kombinasi alfabet B dengan panjang k yang biasa dinotasikan $B^k = 2^k$
   Karena e satu-satu maka $B^k = 2^k$ maka himpunan dari 2^k disebut dengan kode blok (n,k)
   Suatu kode blok linier dengan k = 4 dan n = 7 daigambarkan sebagai berikut :

Tabel Kode blok Biner (7,4)

Kata	Kata Kode
0000	0000000
1100	1011100
0010	1110010
0001	1010001
1001	0111001
0101	0001001
1101	0001101
1010	0011010
0110	1000110
1000	1101000
0100	0110100
1110	0101110
0011	0100011
1011	1001011
0111	0010111
1111	1111111

   Dalam kode blok biner tiap kata kode v panjangnya sama. Karena k<n,n-k bit-bit tambahan ditambahkan pada tiap kata untuk membentuk kata kode. Bit bit tambahan tersebut diharapkan memberikan suatu kode yang mampu mengatasi gangguan pada saluran. Bit bit tambahan tersebut dinamakan digit pariti cek.

2. Kode convolutional
   Misalkan B = {0,1} dan fungsi encoding $e:B^k \rightarrow B^n$ dimana e memetakan u e B^k yang tidak hanya tergantung pada hubungan k bit dari u tetapi juga tergantung pada m bit sebelum u. fungsi encoding seperti itu dikatakan mempunyai orde memori m. himpunan hasil encoding disebut dengan kode konvulutional (n,k,m).
   Contoh : Kode convolutional biner dengan n = 2, k = 1, m = 3. misalkan barisan kata u = (1, 0, 1 ,1, 1, 1) , m bit sebelum u adalah (0, 1, 1) dan barisan pembangkit dari kode tersebut adalah
   $G^{(1)} = (1\ 0\ 1\ 1)$
   $G^{(2)} = (1\ 1\ 1\ 1)$
   Maka kata kode dari u adalah :
   V = (01, 01, 11, 01)

d. Koreksi Kesalahan

Misalkan e fungsi encoding(n,k) dan d fungsi decoding (n,k) yang berpasangan dengan e. Pasangan (e,d) dikatakan dapat mengoreksi k error bilamana v = e(u) dikirim dengan benar atau dengan k error vt hasil pengiriman yang mengakibatkan d(vt) = u.

e. Matriks Generator

Urutan data k bits $d_1, d_2, d_3, \ldots, d_k$ misal dinyatakan dengan vektor d, maka d = $(d_1, d_2, d_3, \ldots, d_k)$. Andaikan kata kode dilambangkan dengan c maka c = $(c_1, c_2, c_3, \ldots, c_k, c_{k+1}, \ldots, c_n)$. Untuk suatu kode sistematis $c_1 = d_1, c_2 = d_2, \ldots, c_k = d_k$. sedangkan

$$c_{k+1} = h_{11}d_1 + \ldots + h_{1k}d_1$$

$$\vdots$$

$$c_n = h_{r1}d_1 + \ldots + h_{rk}d_k$$

dimana :
n = jumlah bit kata kode
k = jumlah bit data asli
r = jumlah bit paritas

Koefisien $h_{ij} \varepsilon$ (0,1) sedangkan $c_{k+1}, \ldots , c_n$ didapatkan melalui operasi mod 2. Pemilihan koefisien tersebut yang menentukan sifat suatu kode. Sebagai contoh lihatlah persamaan cek paritas berikut:

$$\begin{aligned} c_{12} &= d_1 + d_2 + d_3 + d_4 + d_6 + d_8 + d_9 \\ c_{13} &= d_2 + d_3 + d_4 + d_5 + d_7 + d_9 + d_{10} \\ c_{14} &= d_3 + d_4 + d_5 + d_6 + d_8 + d_{10} + d_{11} \\ c_{15} &= d_1 + d_2 + d_3 + d_5 + d_7 + d_8 + d_{11} \end{aligned}$$

maka jika vektor d = (01011101011) 4 bit ceknya adalah $c_{12} = 0$, $c_{13} = 0$, $c_{14} = 0$, $c_{15} = 0$. dan vektor c = dG dimana G haruslah suatu matrks kxn dengan k kolom pertama adalah matriks identitas yang menyatakan jumlah k bit data asli. Sisanya r kolom G menyatakan transpose koefien koefisien h_{ij}.
Sehingga G = [I_k P] dengan

$$P = \begin{bmatrix} h_{11} & h_{12} & . & h_{r1} \\ h_{12} & h_{22} & . & h_{r2} \\ . & . & . & . \\ h_{1k} & h_{2k} & . & h_{rk} \end{bmatrix} \text{ dan G}$$

dinamakan sebagai matriks generator.

$$\text{Untuk contoh diatas P} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \text{ dan}$$

G=

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

f. Matriks Cek Paritas

Untuk dapat mendeteksi sekaligus mengoreksi kesalahan data yang dikirimkan perlu operasi penguraian kode pada penerima.suatu prosedur sederhana adalah mengulang perhitungan cek paritas pengkode dan membandingkan hasil pola cek paritasnya dengan yang diterima. Jika pola yang dihitung dan yang diterima tidak bersesuaian maka pasti ada kesalahan

Andaikan kata kode sistematis c = dG dengan d urutan data k bit, maka c =[d dP] = [d c_p]. Urutan bit cek paritas c_p adalah c_p = dP. Misal c menyatakan suatu urutan kata kode yang diterima dengan jumlah bit nya n. k digit yang pertama menyatakan suatu vektor d yang merupakan bagian dari kata kode yang ditransmisikan. Untuk mengecek apakah ururtan n bits menyatakan suatu kata kode, maka harus dilakukan operasi dP dan membandingkannya dengan urutan cek paritas yang diterima, yaitu c_p

Jika dP + c_p =0 maka urutan yang diterima adalah suatu kata kode yang tepat.

$$dP + c_p = [d \; c_p] \begin{bmatrix} P \\ I_{n-k} \end{bmatrix} = 0$$

Untuk sembarang kode didapatkan c H^T=0

dimana $H^T = \begin{bmatrix} P \\ I_{n-k} \end{bmatrix}$, sehingga

didapatkan H = $\begin{bmatrix} P & I_{n-k} \end{bmatrix}$

dimana H = Matriks cek paritas. Matriks tersebut berperanan sangat penting dalam pembetulan kesalahan.

g. Vektor sindrome

Kata kode harus memenuhi syarat $c\,H^T = 0$. Misalkan suatu kesalahan terjadi dalam digit c maka vektor penerima r dapat dituliskan $r = c + e$ dengan e adalah suatu vektor n bit yang menyatakan kesalahan. Jika kesalahan terjadi dalam bit kedua dan ketiga maka dengan mengoperasikan r dengan H^T akan didapatkan

$$rH^T = (c+e)H^T = eH^T = s$$

dimana s = vektor sindrome. Jika suatu kesalahan terjadi maka s tidak nol

3. METODE HAMMING

Sandi biner yang dipakai harus diubah ke bentuk sandi yang baru dengan cara penambahan paritas. Aturan penambahan paritasnya adalah :

$$2^k \sum_{i=0}^{t} \binom{n}{i} \leq 2^n$$

dengan $\binom{n}{i} = \dfrac{n!}{i!(n-i)!}$

dimana :
- k = jumlah bits data
- n = jumlah bits data yang sudah ditambah paritasnya
- t = jumlah bits error yang mungkin dideteksi
- r = n – k = jumlah bits paritas

a. Pelacakan Kesalahan

Pada umumnya penanganan kesalahan dalam pengiriman data dilakukan dengan mengulang kembali pengiriman data. Bila menggunakan metode hamming persoalan tersebut dapat diatasi dengan koreksi kesalahan secara otomatis dengan menggunakan sandi tertentu. Dengan metode tersebut pesan akan diubah menjadi angka 0 dan 1. agar kesalahan yang mungkin terjadi dapat dikoreksi maka penerima perlu mensandikannya kembali.

Untuk itu diperlukan suatu matriks H dan T dimana :

- Matrik H: Matrik yang terdiri dari r kolom matrik diagonal dan n kolom matrik sembarang yang elemen-elemenya 0 atau 1, dengan n adalah cacah digit sandi digital yang akan dikirimkan,sedangkan r adalah bit paritas (bit tambahan)
- Matrik T: Matrik yang elemen-elemennya merupakan sandi digital yang akan dikirimkan.

Untuk itu dipilih matrik H yang menghasilkan H.T=0. Pada penerima, isyarat yang diterima dimisalkan sebagai matrik R, dikalikan kembali dengan matrik H dan menghasilkan isyarat sindrom S.

Bila S=H.R=0, berarti isyarat yang diterima sudah benar atau cocok dengan isyarat yang dikirimkan. Tetapi jika S=H.R≠0, berarti isyarat yang diterima ada kesalahan. Kesalahan yang terjadi bisa dilihat dari isyarat sindrom yang terbentuk, dengan mencocokkan isyarat sindrom dengan matrik H akan dapat diketahui kesalahan yang terjadi pada angka ke berapa. Sebagai contoh, jika isyarat sindrom cocok dengan kolom ke 5 dari matrik H, berarti kesalahan terjadi pada angka ke 5 dari pesan yang dikirimkan.

Matrik H bias dipilih sembarang, dengan ketentuan tidak boleh ada kolom yang mempunyai elemen-elemen persis sama. Dengan alasan inilah, maka matrik H dipilih sebagai berikut.

$$H = \left[\begin{array}{ccccccc|cccc} \multicolumn{7}{c}{\overbrace{\qquad\qquad\qquad\qquad}^{n\ \text{kolom}}} & & & & \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & & & & & & \multicolumn{4}{c}{\underbrace{\qquad\qquad}_{r\ \text{kolom}}} \end{array}\right]$$

b. Pembentukan Sandi Baru (Encoding)

Misal akan dicari sandi baru untuk pesan A yang mempunyai sandi asli 1000001. Perkalian matrik H dengan vector T yang mempunyai 7 elemen pertama sama dengan sandi lama yang akan diubah dan 4 berikutnya adalah elemen yang akan dicari nilainya, dapat dinyatakan sebai berikut:

$$H.T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

dari hasil perkalian diatas diperoleh nilai

$c_1 = 1$ $\quad$ $c_3 = 0$

$c_2 = 0$ $\quad$ $c_4 = 0$

Sandi baru diperoleh dengan menggabungkan sandi asli dengan 4 elemen baru yang diperoleh dari perhitungan diatas. Dengan demikian sandi baru untuk pesan A adalah 10000011000.

c. Deteksi dan Koreksi Kesalahan

Misalkan dikirim suatu pesan yang oleh penerima pesan tersebut diterima sebagai sandi 10000011001. Untuk melihat apakah pesan ini benar atau tidak, maka pesan yang diterima tersebut harus dicek.

Untuk mencek sandi yang diterima, perlu dicari isyarat sindrom, yaitu perkalian antara matrik H dengan sandi yang diterima. Hasil perkaliannya adalah

$$H.R = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Isyarat sindrom yang diperoleh dari hasil perhitungan diatas adalah $[0\ 0\ 0\ 1]_{-1}$. Jika isyarat sindrom ini dicocokkan dengan matrik H, terlihat bahwa isyarat sindrom cocok dengan kolom ke 11. Dengan demikian, kesalahan terjadi pada angka ke 11, yaitu dari angka "1" harus diubah menjadi angka "0".

4. PROSES

4.1. Proses Pembentukan Sandi baru (Encoding)

Untuk pembentukan sandi baru diperlukan urutan urutan seperti pada gambar 1.

a. Deteksi dan Koreksi Kesalahan

Algoritma untuk mendeteksi dan mengkoreksi kesalahan bisa dilihat pada gambar 2.

4.1.1. Proses Pesan Salah

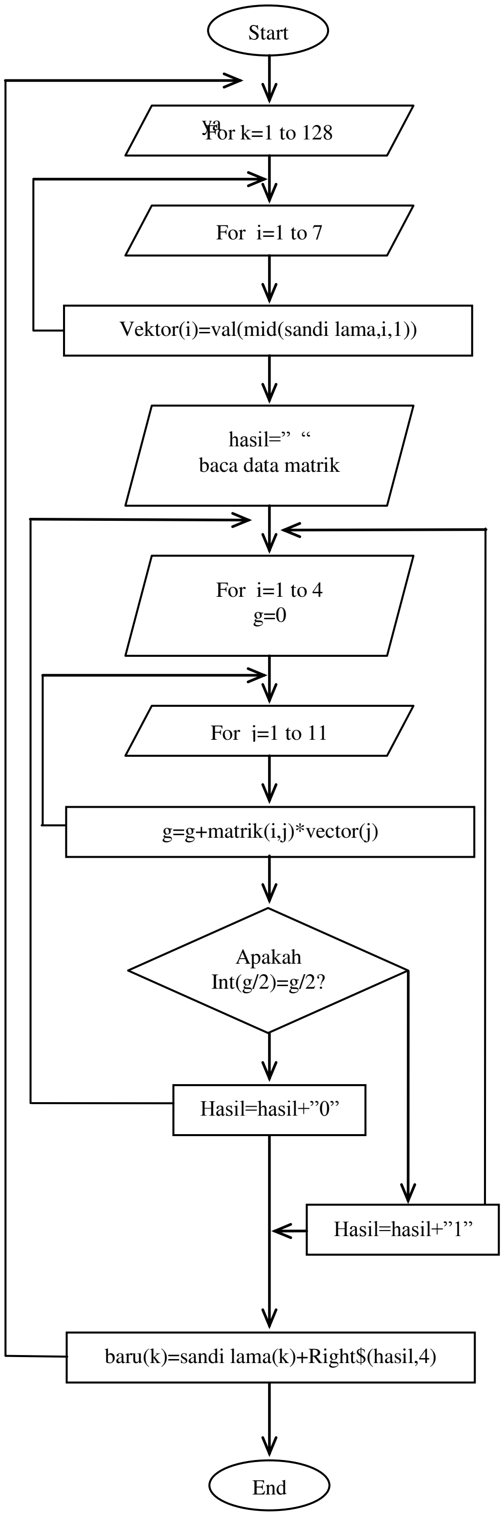
Algoritma untuk memproses pesan kesalahan bisa dilihat pada gambar 3.

b. Proses Pembentukan vector Sindrom

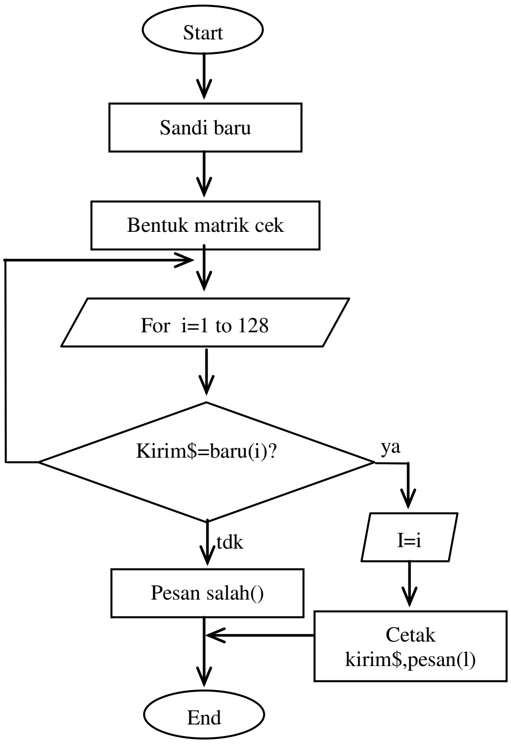
Algoritma untuk memproses pesan kesalahan bisa dilihat pada gambar 4.

c. Proses Pembetulan

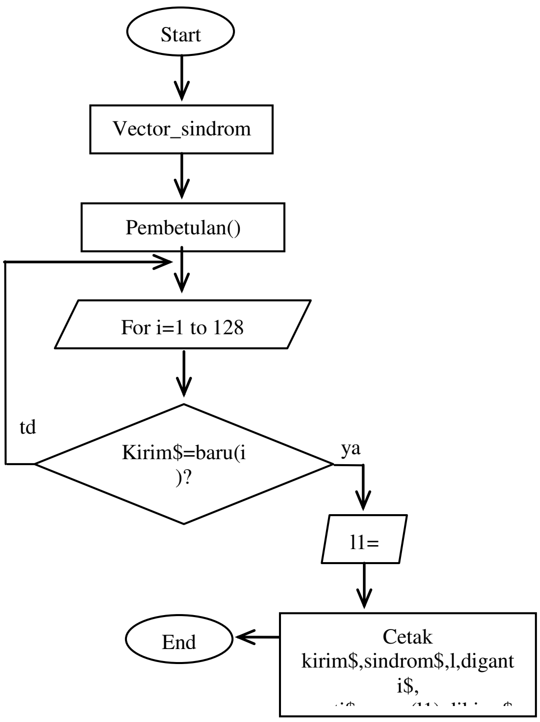
Algoritma untuk pembetulan bisa dilihat pada gambar 5.



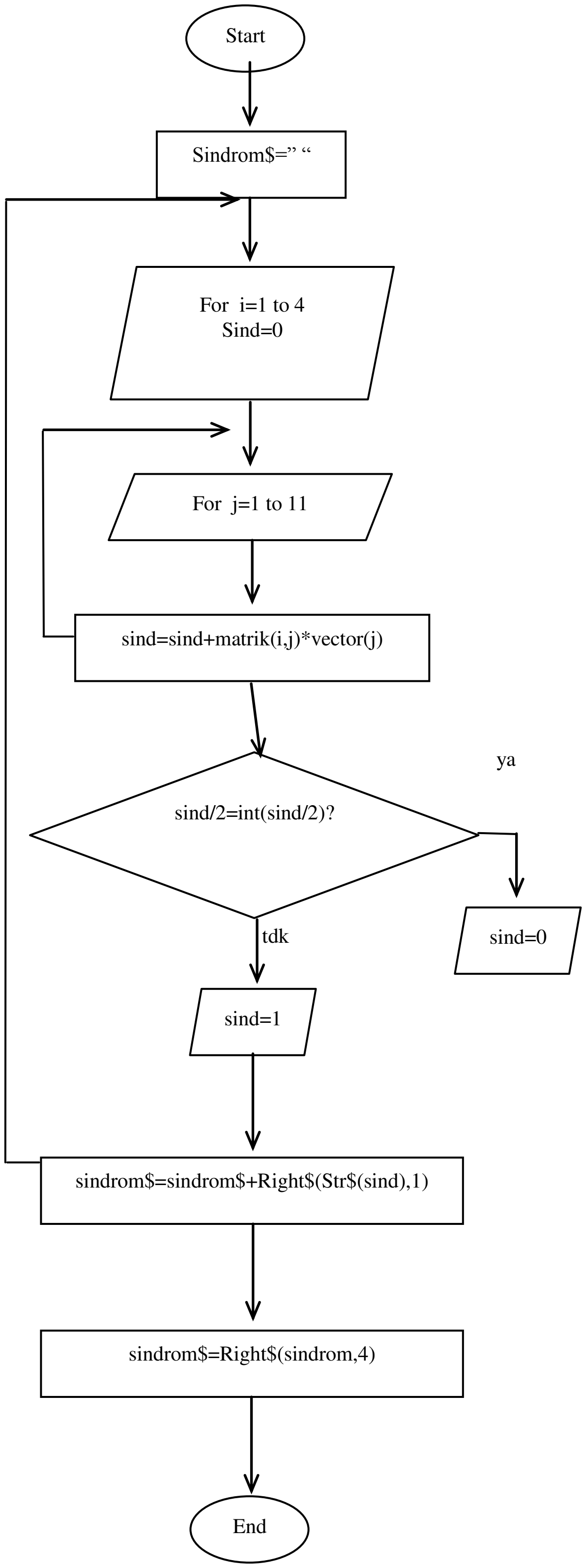
Gambar 1. Flowchart pembentukan sandi baru



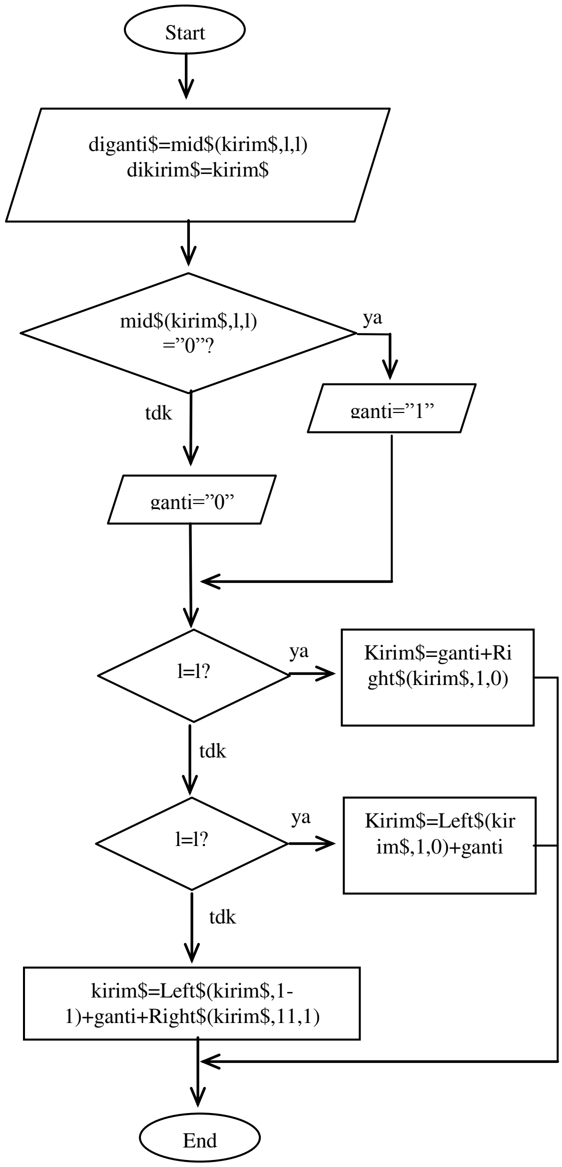
Gambar 2. Flowchart untuk deteksi dan koreksi kesalahan



Gambar 3. Flowchart untuk memproses pesan Kesalahan



Gambar 4. Flowchart untuk membentuk vector syndrome



Gambar 5. Flowchart untuk pembetulan

5. KESIMPULAN

Suatu kata yang dikirim oleh seseorang melalui saluran pengiriman yang mendapat gangguan (noise) akan mengakibatkan kata yang diterima mungkin berbeda dengan kata yang dikirim

Cara yang diterapkan untuk mengatasi akibat gangguan tersebut adalah dengan menambah digit-digit tambahan pada data yang dikirim sedemikian rupa agar nantinya kata yang diterima akan sama dengan kata yang dikirim.

Penambahan jumlah bit paritas tergantung dari bit error yang ingin dikoreksi. Bit paritas dapat dibuang dari bit data aslinyasetelah proses pengoreksian selesai.

6. DAFTAR PUSTAKA

1. MAN Young RHEE, "Error Corecting Coding Theory", Mc Graw Hill, 1999.
2. K Sam Shammugam, "Digital and Analog Communication System", John Wiley and Son, 1999.
3. J Andrew Viterbi dan Omoda K Jim, "Principals of Digital Communication and Coding", Mc Graw Hill, 1998.